M-LEARNING - A MULTI AGENT APPROACH

Terje Solsvik Kristensen and Joachim Coetzee Kjærvik
Western Norway University of Applied Sciences,
Department of Computing, Mathematics and Physics, Bergen, Norway

ABSTRACT

In this paper we have developed a framework for learning by mobile based on JADE and JESS as software components. Such a software integration makes it possible to exhibit ACL communication properties from JADE and rule-based reasoning capability from JESS. Software agents are implemented following the guidelines of the theoretical concept of Zone of Proximal Development (ZPD), as a means of supporting and promoting opportunities for effective learning.

KEYWORDS

M-Learning, MAS, JADE, JESS, Intelligent Tutoring, Zone of Proximal Development (ZDP)

1. INTRODUCTION

Education is a domain to which agent technology has frequently being applied. This area has been gaining more attention for instance research on artificial intelligence applied to computer assisted learning [Ayala, Yano, 1998], [Jennings, Sycara, Wooldridge, 1998], [Hay, Kichin, 2004], [Novak, Canâ, 2006/2008]. In classroom learning one is mostly based on behavioral learning theories where the learner is the object of assessment. The teacher initiates the learning process and the learner responds. Another learning approach, constructivism, focuses on the learner’s abilities to develop her own mental models and learning concepts [Kichin, 2000], [Kinney, Georgeff, Rao, 1996]. This approach has more and more become accepted to be a more relevant method to promote learning, even at the university level.

Agents often have a repertoire of available actions that can manipulate their environment. This set of possible actions is what constitutes an agent’s actuators. For agents to be able to operate independently and represent our best interests, we delegate high-level objectives, by means of some deliberation logic. This logic is what constitutes an agent’s decision-making capability. An agent’s decision-making strategy is dictated by the information it has available in its Knowledge Base (KB) and its high-level objectives.

2. AGENT ARCHITECTURES

The implementation of an agent can be done in a number of ways. An agent is typically characterized by its internal architecture or agent model. Wooldridge [2000] defines agent architectures as “the key problem facing an agent is that of deciding which of its actions it should perform in order to best satisfy its design objectives. Agent architectures are software architectures for decision-making systems that are embedded in an environment”.

2.1 Belief-Desire-Intention

When describing complex systems, using abstractions, often makes it easier to explain and grasp technical concepts. Shoham [1997] describes agents using mentalistic states of beliefs, desires (choices), and intentions (commitments):

“An agent is an entity whose state is viewed as consisting of mental components such as beliefs, capabilities, choices, and commitments“.
Talking about agents as if they have mental state to describe and predict their behaviour is a way of letting abstracting concepts be closer to human experiences. Singh [1998] describes the mental states of an agent as beliefs, characterizing what an agent imagines its world state to be, the goals that describe what states the agent would prefer and desires that describes the agent’s preferences. The BDI model [Kinney, Hay, Adams, 2000] is an abstraction that lets us predict and explain how agents behave. The assumption is that by attributing agents with beliefs and desires, we expect them to act rationally in order to try to accomplish their desires, given their beliefs. By using this model, we may express reasonably briefly the behaviour of a system, without going into too low-level details e.g., at the physical or the design level.

2.2 Type-Message Agents

Charles Petrie [1996] discusses one specific class of agents so-called typed-message agents. The main difference between typed-message agents and other classes of agents is their ability to communicate using shared message protocols.

This class of agent is defined in terms of agent communities that mainly exhibits three properties: accomplishing tasks in a system requires a community of agents to exchange messages. The communication is based on using a shared message protocol where some of the message semantics are independent of the application and type. The semantics of the message protocol require a transport protocol.

2.3 Deductive Reasoning Agents

When building artificially intelligent systems, symbolic AI has traditionally been a commonly used technique. A system that exhibits intelligent behaviour, according to symbolic AI, is a system that should have a symbolic representation of its environment in which it operates and a symbolic representation of its desired behaviour, syntactically manipulating the representation.

In deductive reasoning agents’ symbolic representations are logical formulae and the act of manipulation equates to logical deduction. Agents based on such an architecture typically embed a knowledge base KB, containing formulae in First-Order Predicate Logic (FOPL) [Russel, Norvig, 2000] that are used to represent properties of an agent’s environment.

An agent’s KB is the actual information it has available of its environment, and can be thought of how we humans have beliefs. For instance, a person might believe that valve 221 is open, whereas an agent might have the predicate Open (valve 221) in its KB. However, it is important to note that just like humans, agents can be wrong. Deduction rules are essentially rules of logical inference that agents use to derive new information. By taking perceptual information as input, an agent will use its deduction rules to attempt to derive optimal actions, given the situation.

3. AGENT COMMUNICATION LANGUAGES

In order to communicate the agents should use some form of agent communication language (ACL). The language ACL is responsible for defining the types of messages that agents are able to exchange in the system. In addition, the semantics of these messages have to be shared and known before runtime. In other words, the shared message protocols discussed in typed-message agents, are what constitute an ACL. By using these protocols and having messages with defined semantics, agents are able to understand and converse with each other in a common language. The actual communication using an ACL is handled by lower-level protocols such as for instance Simple Mail Transfer Protocol and Transmission Control Protocol (TCP)/Internet Protocol (IP).

3.1 Speech Acts

In a general scenario, an agent would be unable to force another agent to perform some action or write information onto its internal state. What agents do, performed as actions or Communicative actions (CA) – is an attempt to influence other agents [Austin, 1962]. A simple example would for instance be stating the fact
"It is raining in Bergen”. Under normal circumstances, stating such a CA would be an attempt to modifying your beliefs. However, by simply stating or uttering the fact alone would usually not be enough to convince you to change your beliefs - you have control over your own beliefs (desires and intentions). Upon stating this fact it can be thought of as an attempt to change someone’s internal state. In addition, since the utterance is an action in itself - it is performed with some purpose or intent. Thus, the assumption is to make someone believe that it is raining.

This concept is drawn from Speech Act Theory which has heavily influenced the development of many of today’s ACLs. In his book, Searle [1969] describes Speech Act Theory as, "Speech act theory treats communication as actions. It is predicated on the assumption that speech actions are performed by agents just like other actions, in the furtherance of intentions.”

4. THE FIPA COMMUNICATION LANGUAGE

FIPA’s Agent Communication Language (ACL) is based on speech act theory, whereby messages are actions or communicative acts that are intended to perform some action. The FIPA ACL specification is composed of a set of message types and the description of their effects on the mental attitudes (BDI) of the sender and receiver agents. This ACL defines an outer language for its messages, whereby each ACL message contains a message parameter – the performative. The performative denotes a message’s type of communicative act, which in turn defines the actual intended meaning of a message. The FIPA Communicative Act Library (CAL) also defines an inner, or content language that is used to describe the content of a message. This includes how the content is expressed, how the expression is encoded in the message, and how a receiving agent should interpret the expression. This allows agents to “understand” each other, such that they ascribe a shared meaning of the content. It is important to note however, that FIPA ACL does not dictate any specific language for the content of a message.

4.1 Semantics of the FIPA ACL specification

A Semantic Language (SL) is the formal language used to define FIPA ACL’s semantics. This language makes it possible to represent beliefs (B), desires (D), uncertain beliefs (U), and intentions of agents (I) (persistent goals (PG)), as well as actions the agents can perform. The semantics of each communicative act is specified as a set of SL-formulae that describes the communicative act’s Feasibility Precondition (FP) and its Rational Effect (RE). The FP denotes a constraint that a sender agent must satisfy to be considered conforming to the specification. The RE of an action denotes the intent of a message, meaning what an agent wants to achieve by sending the message (perlocutionary act). It is important to note however, in a MAS consisting of autonomous agents, the RE of a message is not guaranteed. This means that conformance to the standard does not require the receiver of a message to respect the RE part of the ACL semantic - only the FP part.

5. LEARNING BY MOBILE PHONE

There is a lot of excitement about mobile technology today. Mobile devices are being used in an increasingly wide variety of ways. With the ubiquity of smart devices and the processing power they provide a phone no longer is just a phone. The phones have essentially become computing devices in their own right. We use them for banking, sending emails, consuming media and learning.

Learning by a mobile or m-learning we may call it, is the term referring to learning using mobile devices, and can be defined as learning mediated through mobile technologies. m-learning can be thought of as the natural evolution of e-learning [Kristensen et al., 2006-2013] whereby learning material is delivered on mobile platforms rather than on computers. Learning anytime anywhere or “ubiquitous learning” is just one of the many reasons why mobile learning has spurred on a lot of interest. By using mobile devices, one is not physically constrained to any specific location. In addition, mobile devices have at a steady rate grown in terms of processing power and memory capability. This trend is partly due to the continual reduction in cost
of computing capability and the devices are becoming more powerful and more reasonable in price. By having more power available, we are able to do interesting things that a couple of years ago may have been unfeasible. Thus, bringing “heavy” paradigms such as agent technology to mobile platforms is more viable now than ever.

5.1 Learner Modelling

Learner modelling is a concept drawn from artificial intelligence. It is the key concept concerning the application of artificial intelligence in educational software. Learner models are important within computer-based systems intended to promote learning because they provide the means to support individually adapted instruction. The task of ‘learner modelling’ or ‘cognitive diagnosis’ (that is, the task of building a learner model) is the process of inferring the learner’s knowledge by analyzing his or her behaviour.”

A learner model is in other words, a system’s derived representation of a learner’s abilities. This representation is based on the information a system is able to obtain from its learner. Learner models are typically used in decision-making systems in an attempt to promote ‘effective learning’. By using these models, autonomous agents could for instance be able to adapt to a learner’s individual needs. This would make it possible to differentiate the material that is presented to each learner, and the agent could attempt to derive optimal tasks to be working on, based on the information it has available.

In the individualization approach of learner modelling, the effectiveness of a system is quantified based on its ability to dynamically adapt to each individual learner’s needs. In order for this adaptation to work, it relies on a “student model” or a learner model. In the group adaptation approach a system will adapt to groups of learners. This necessitates a representation of groups rather than individual learners – a group model.

There are a number of different ways to implement learner models, one concrete instance of this technique used in an agent-based context, was presented in the paper by Ayala and Yano [1998]. Here the learner model is implemented in terms of the theoretical concept of Zone of Proximal Development, presented by the Russian psychologist L. Vygotsky [1995].

5.2 Zone of Proximal Development

The concept of Zone of Proximal Development (ZPD) is based on “the premise” of two development levels in a learner’s learning process. The Actual Development Level (ADL), comprising of those knowledge elements (belonging to some domain) that have been internalized by the learner as a result of a completed development cycle. The Potential Development Level (PDL), comprising of those knowledge elements that the learner constructs in collaboration with other learners or with the assistance of a more knowledgeable others (MKO). Vygotsky defines the ZPD as “the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers.”

![Figure 1. Zone of Proximal Development](image)

The basic idea of this theory is that learning enhances a learner’s developmental level. In order to learn we must be presented with tasks that are just slightly beyond our current reach of abilities. In other words, solving the tasks that are too complex or too simple, does according to this theory, not promote learning. The tasks that are just slightly beyond a learner’s present abilities exist in the ZPD. These are tasks that a learner
is almost able to solve independently, but require some degree of guidance. After having received help on a particular task, the idea is that learners will eventually be able to do it themselves – bridging the gap between the ADL and the PDL. By presenting tasks that fall within a learner’s ZPD, we attempt to promote effective learning (https://www.researchgate.net/publication/258519369_Effective_Learning_in_Classrooms ). One important thing to note with this theory, is that every learner will have his own ZPD.

5.3 Agents in a Mobile Learning Environment

Based on the definition and description of ZPD, it can be derived that there is a dynamic aspect to learning using this approach. Using this model, a system attempts to promote effective learning by attempting to adapt and identify the optimal tasks each individual user should be working on. The ZPD is in no way static, it constantly shifts. As a user develops new abilities, the ZPD will progressively move further or become wider in scope.

A mobile learning environment based on this theory will have certain computational requirements that demand flexibility, autonomy, and social behaviour. These are all aspects that agent technology may provide. This adaptive learning approach will involve an agent using a learner model to determine which tasks would be most beneficial for a given user in a given situation. The decision-making or reasoning would be based on what a user has been able to internalize in his knowledge domain.

6. JADE AND JESS

In JADE the software agent abstraction is implemented as a Java thread, one thread per agent. Each running instance of JADE runtime is called a container [Shenghuan, Kungas, Matshin, 2006], [Naser, 2008]. By using FIPA ACL to represent messages, the agents are able to interact using the asynchronous message passing mechanism providing by JADE [Bellifemini, Caire, Greenwood, 2007]. In order to endow agents with the capability to reason, it is possible to integrate external software with the Behaviour class. One software suite that is compatible with JADE is JESS [Friedman-Hill, 2000]. JESS is an expert system shell that supports a rule-based symbolic language, similar to CLIPS. JESS is described as from the official homepage given by (http://jade.atilab.com/support/faq/ ). JESS is small, light and one of the fastest rule engines available. JESS also provides a scripting environment. The rule engine that JESS implements is based on an enhanced version of the Rete algorithm [Forgy, 1982] for processing rules.

By integrating JESS with JADE it is possible to implement declarative decision-making modules. These modules enable agents to reason with respect to the declarative language. The agents are therefore rule-based JADE-agents that are capable of doing automated reasoning, making them more “intelligent”. The Rete.run() method allows us to run the inference engine and will make the engine fire all rules that apply to facts in the current working memory. The engine returns once there are no more rules to fire.

7. CONCLUSION AND FURTHER WORK

In this paper we have laid the foundation of a framework of a system for learning by the mobile phone. The implementation is based on two separate software components JADE and JESS. This software integration has resulted in an architecture that exhibits ACL communication properties derived from JADE, and rule-based reasoning capability derived from JESS. A learner model has additionally been implemented in an attempt to facilitate learning optimization or ‘effectively learning’.

Vygotsky’s theoretical concept of ZPD serves as the foundation of the agents’ behaviour. By using the ZPD as a guideline, the agents attempt to promote learning by deriving optimal tasks and proposing them for their users. However, this requires that the agents are able to collect information about the user and inferring a representation of their abilities. The concept of Learner Modelling supports such a process. The learner model is used in the agent’s decision-making process and enables the agents to propose tasks that are specific to each individual user’s ability level.
REFERENCES


